Amendments to the Claims:

1.-12. (canceled)

13. (currently amended) A system for generating automation code for a manufacturing and/or processing plant from a description enriched with control-relevant information, the system comprising:

~~components in the description described in~~ a description comprising a drawing showing a layout of components of the plant based on a material flow in the manufacturing and/or processing plant, wherein the drawing ~~comprises~~ shows ports with control-relevant information for each component, and ~~the components have ports and are represented by~~ the drawing shows at least one functional module for each component, wherein

input/output information is mapped to the ports, wherein the input/output information stems from directed relationships between the components, wherein the input/output information comprising predecessor/successor relationships among the components is included in the description, wherein

signals provided for a transmission via the ports of the components are associated with ~~the~~ each functional module and further comprising:

a first mechanism for defining metainformation for the signals; and

a code generator for generating automation code by interconnecting the signals, wherein the automation code is generated on the basis of a structure of the plant and know-how, including the predecessor/successor relationships, previously input into the description.

14-16. (canceled)

17. (previously presented) The system according to claim 13, further comprising a mechanism for inputting control-relevant information for use in the description.

18. (canceled)

19. (currently amended) The system according to claim 13, wherein the material flow, and/or an energy flow, and/or an information flow in the plant is provided as a basis for mapping the ~~directed~~ predecessor/successor relationships between the components.

20-22. (canceled)

23. (previously presented) The system according to claim 13, wherein the generation of automation code is provided for central and/or distributed automation solutions.

24-25. (canceled)

26. (currently amended) A method for generating automation code for operating controllers in a manufacturing and/or processing plant from at least one description enriched with control-relevant information, the method comprising:

creating a description comprising a drawing of a layout of the plant, the layout representing components ~~described in the descriptions~~of the plant by at least one respective functional block or building block per component ~~in a drawing~~ based on a material flow in the plant, wherein the drawing comprises control-relevant information, and shows at least one port for each component ~~has at least one port~~;

mapping input/output information regarding the ports between the components, wherein the input/output information stems from directed relationships including predecessor/successor relationships among the components contained in the descriptions;

~~transmitting~~ defining signals associated with the functional blocks or building blocks via the ports of the components;

defining metainformation for the signals; and

generating automation code in a code generator for operating the controllers by interconnecting the signals, wherein the automation code is generated on the basis of a structure of the plant and know-how, including the predecessor/successor relationships, previously input into the description.

27-30. (canceled)

31. (previously presented) The method according to claim 26, wherein automation code is generated for central and/or distributed automation systems.

32. (canceled)

33. (currently amended) A system for generating automation code for a manufacturing and/or processing plant, the system comprising:

a plant description comprising a plurality of components, each component representing a given element of the plant, each component comprising at least one function module and at least one port, each port representing a connection point on the given element for data communication with another element of the plant, each function module being a reusable software object type that defines characteristics and functions of the given element;

a communication network within the plant comprising a respective controller connected to each of the plant elements;

the description comprising a drawing showing a layout of the components based on a flow of material in the plant ~~and~~, the description further comprising control-relevant information comprising rules that specify all allowable relationships including predecessor/successor relationships among the plant elements, including allowable information content and flow directions among the ports; and

a code generator that automatically generates automation code for the plant that controls information flows among the controllers based on the drawing and the control-relevant information of the description, wherein the automation code is generated on the basis of a structure of the plant and know-how, including the predecessor/successor relationships, previously input into the description.

34. (currently amended) The system of claim 33, wherein the network comprises at least two control zones, each control zone comprising a subset of the plant elements controlled by a respective subset of the controllers, and the network further comprises a coordinating controller for each control zone, and wherein the description describes a topology of the network for the automatic code generation.

35. (new) The method according to claim 26, wherein the metainformation comprises one or more input/output parameters with a value "S" or "P" for each component, and wherein an algorithm operates the code generator to automatically generate code for connecting the components as follows:

for all components

for all inputs of the respective functional module

for all predecessors of the component

a) search for a predecessor functional module that has an output parameter with a value "S";

b) search for an input of the respective functional module that has a parameter with a value "P"; and

c) connect the output of the predecessor functional module that has an output parameter with a value "S" to the input of the respective functional module that has the parameter with a value "P".